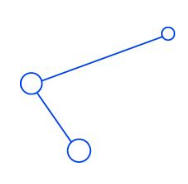
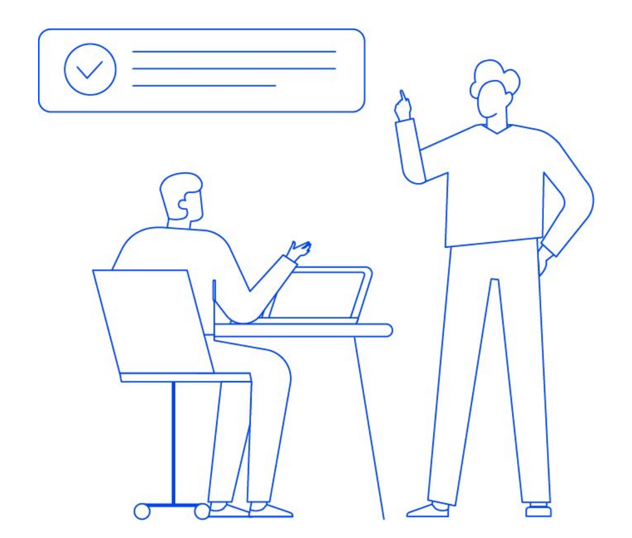


Переход с TCP на UDP
в доставке видео
HL
HighLoad++
2022

О чем доклад?



1. Раннее развитие онлайн видео было на TCP
2. Сегодня индустрия доросла до возврата на UDP



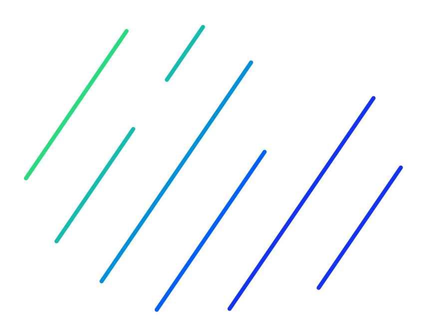


Что было до онлайн (<2008)?

- UDP MPEG-TS
- RTP

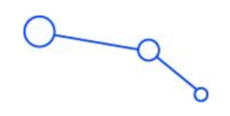
Что примечательно:

- RFC4588 2006 год, retransmit
- RFC5109 2007 год, FEC



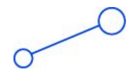


Борьба с потерями



С доисторических времен люди умели бороться с потерями в UDP:

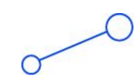
- Избыточная посылка
- Повторная посылка



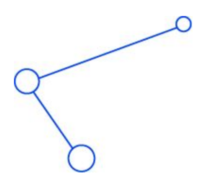




С чего началось онлайн видео?



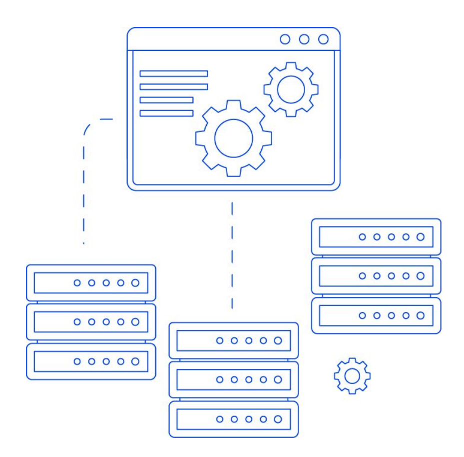
- По объёму влияния на индустрию — RTMP + Adobe Flash.
- Кризис 2008 года высвободил оффлайн деньги для онлайна.
- Всё это жило на TCP.
- Adobe не осилил поднять RTMFP.
- Плагины тоже не получилось сделать хорошо.





Чем хорош TCP для видео?

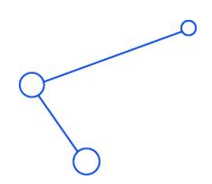
- Ретрансмиты из коробки.
- Эффективная реализация в сетевом стеке.



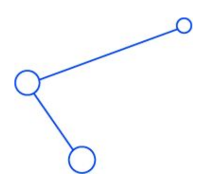


Как выглядит раздача 40 Гбит/с?

- Записали в каждый сокет по 100 килобайт.
- Уснули в реакторе до готовности следующего.
- Работает ядро и сетевая карта.



Чем плох TCP для live-видео?



- Head-of-line blocking.
- Backpressure и congestion control делают больно.
- Пришлось изобретать libevent для эффективной работы.
- Невозможность пропуска данных.

Прочие ограничения типа:

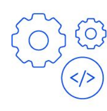
- Нехватка сигнализации отправки (сложно дропать кадры).
- Невозможность мигрировать соединение между серверами.
- Отсутствие multipath tcp.

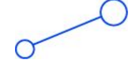


Head-of-line blocking

- Альфа и омега переезда на TCP.
- Проблема больше для HTML, чем для видео.



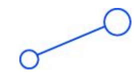




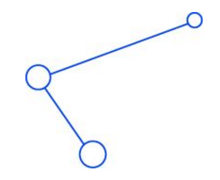




Backpressure + CC

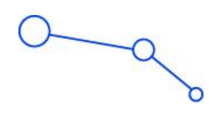


- Для live-видео не требуется вообще.
- Мешает заполнять буфер плеера.
- Требует «разгона» соединения.
- Умеет снижать скорость передачи без реальных причин.
- Усложняет серверный код.

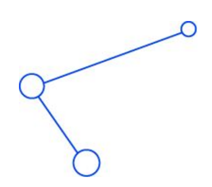




Libevent



- У нас две копии шедулера: на 50 и на 50 000 процессов.
- Вопрос с рваным потоком выполнения не закрыт (ловля исключений).
- Попытки таскать туда-сюда tcp и протокольный стек.



flussonic

Программисты привыкли к TCP

- 20 лет пишем всякие реакторы для TCP: c10k, c100k, c1M.
- Спрайты, бандлы и прочая упаковка.
- Keepalive, reconnect.





Нехватка сигнализации отправки

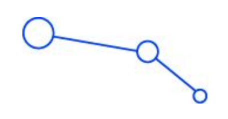
- Неизвестно что уже пришло на ту сторону.
- Ядро не сообщает, что отправило на ту сторону.
- Есть информация только о том, что данные скопированы в ядро.

И что?

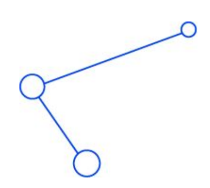
- Нужна сигнализация доставки на уровне протокола.
- Сложно мелкогранулярно пропускать отдельные кадры.
- Баланс между большим буфером отправки и возможностью пропуска.



Привязка состояния к серверу



- UDP соединение можно перенести на другой сервер.
- TCP соединение смигрировать практически невозможно.
- Мы привыкли жить с рвущимся TCP.
- UDP и рвать не надо — и это совершенно непривычно.



Как решали 10 лет назад?

- Перешли на HLS.
- Соединение стало неперсистентным.
- Гранулярность потока выросла до секунд.
- Цена: задержка до минуты (соседи орут ура, а ты ещё не знаешь).

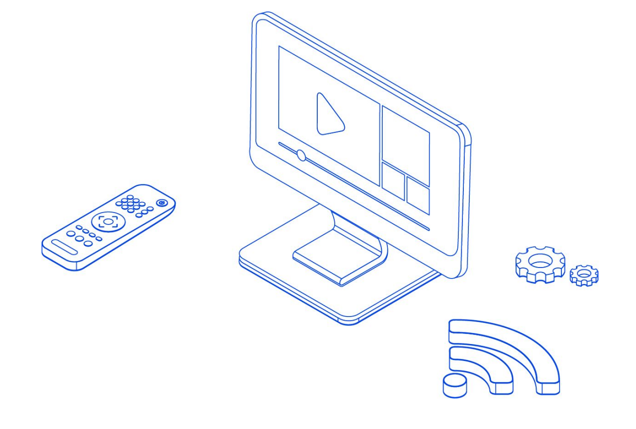


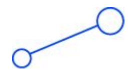
Что поменялось?

- Спорт
- Realtime медиа
- VR/AR
- Все хотят стабильно низкой задержки

Что поменялось?

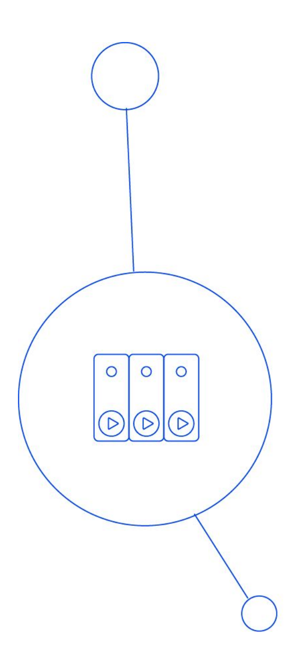
- Спорт
- Realtime медиа
- VR/AR
- Все хотят стабильно низкой задержки.

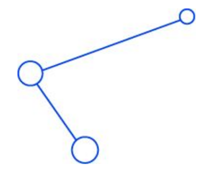




Что предлагает TCP доставка?

1. Уменьшение длины чанков.
2. Серверный пуш.
3. По сути возврат к RTMP с новыми деталями.

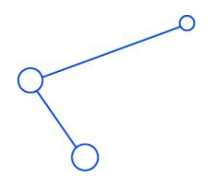




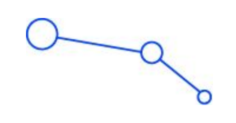
flussonic

Что предлагает UDP доставка?

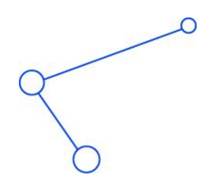
1. Вне браузеров всё есть и очень давно.
2. В браузерах WebRTC и WebTransport.
3. Решение имеющихся проблем и создание новых.



Как оно развивалось вне браузеров?

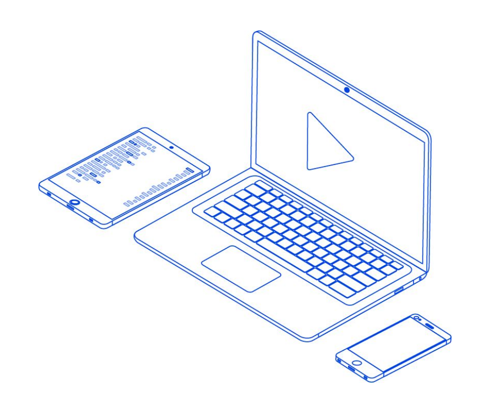


- Масса проприетарных решений: Octoshape, Live-u, Zixi и т.п.
- MPEG-TS эволюционировал в открытый SRT.
- Самый развитый — WebRTC.

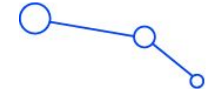


WebRTC

- Вырос из видеоконференций.
- Сегодня активно внедряется для онлайн-видео.



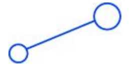


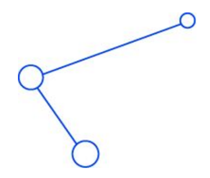




Чем хорош WebRTC

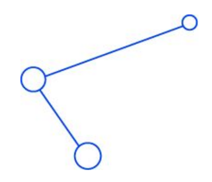
- Умеет смиряться с потерями.
- Стабильная задержка, нет плавающей буферизации.



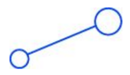


Как компенсируются потери?

1. Встроенные ретрансмиты.
2. Есть буферизация для реордеринга и ретрансмитов.
3. FEC на уровне RTP (для видео) и в аудиокодеке Opus.
4. Дублирование пакетов (RED).



WebRTC и TV



- Массовая доставка требует ABR.
- Как измерять скорость, если нет чанков?
- Гипотеза о буферизации пакетов и сжатии IAT.





Проблемы WebRTC

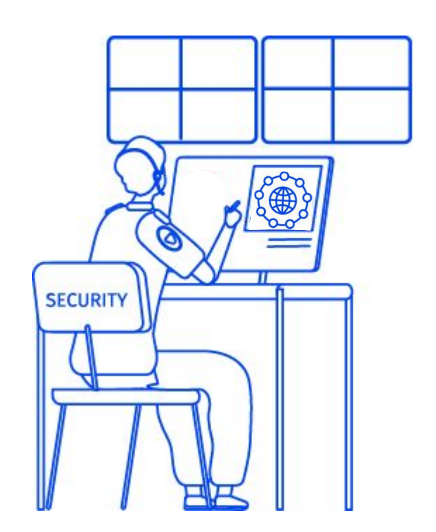
1. В браузерах у MSE и WebRTC разные декодеры.
2. Нет DRM.
3. По сути нельзя гонять premium контент.
4. Не было протокола для публикации/проигрывания.

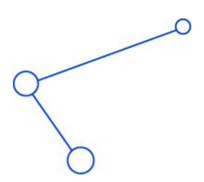
Всё это не UDP проблемы



Безопасность и WebRTC

- Безопасники не любят утечку IP адресов.
- Возможность локальной связи браузеров не всем нравится.





Сетевая сложность

- WebRTC внутри на UDP имеет двухстороннее соединение.
- Весь интернет строился вокруг направленного TCP.
- Все шлюзы, роутеры, файрволы не готовы.

how to filter tcp connection

All Videos Images

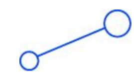
About 23,300,000 results (0.49 sec

how to filter udp connection

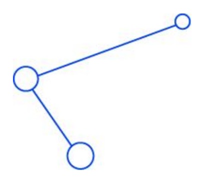
All Videos Images

About 7,800,000 results (0.49 sec

WebTransport



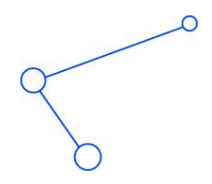
- https://web.dev/webtransport/
- HTTP/3 (udp) аналог вебсокетов.
- Ноябрь 2019 IETF quic datagram RFC9221.
- Первые прототипы webcodecs.
- Попытка сделать webrtc на javascript.





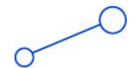
Проблемы WebTransport

1. HTTP/3 очень сложный (идея в том, чтобы выключить CC).
2. Стабильных реализаций нет, всё меняется слишком быстро.
3. «Очень перспективная» инициатива.



Потенциал webtransport + webcodecs

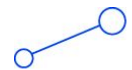
1. Новые кодеки.
2. Возможное внедрение SVC.
3. Бурное и разнообразное развитие на базе wasm.





Точки развития вокруг UDP

- SVC
- Комбинирование SVC, FEC, RED, RTX.
- Отказ от ретрансмита в WiFi.
- Что-то ещё.



flussonic

Вопросы?

Максим Лапшин

max@maxidoors.ru

flussonic

